

# Complexity & Algorithms, Spring '26

→ Matchings  
→ Class recap



# What's left?

- For me:
  - grading essay + final HW
  - finishing last few regrades
  - syncing scores to Canvas
  - writing a final!
  - **office hours next Monday**
- For you:
  - Oral HW: by Thursday:  
everyone signed up?
  - Study: practice exam  
+ extra problems posted

# Final topics

## First half:

- Recursion
- DP
- Greed + approximation
- NP-Hardness + reductions

emphasize 2<sup>nd</sup> half  
↓

## Second half!

- Graph algorithms:
  - BFS / DFS
  - Top sort (+DP)
  - Shortest Paths
- LPs
- Lower bounds
- FFTs
- Max Flow + applications

# Final exam format:

- 7 questions

- I'll take 5 highest scores

- Cheat sheets - hand written -  
are allowed

# Last topic: Flows + matchings

We saw max flow used to find matchings in bipartite graphs:

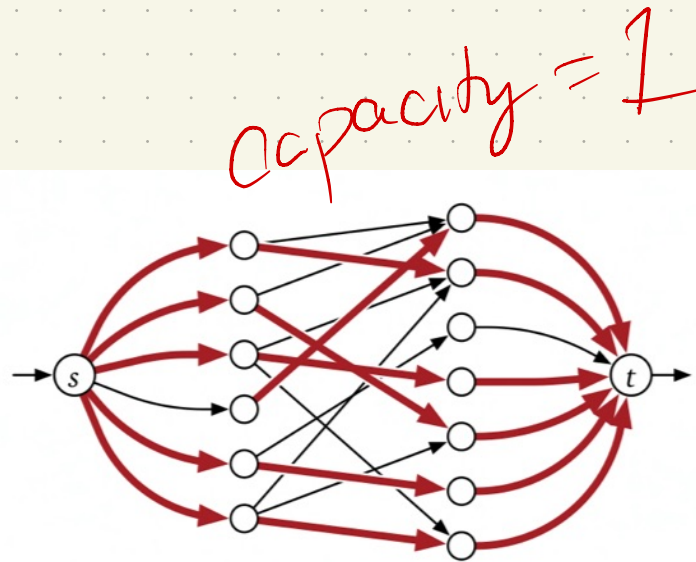
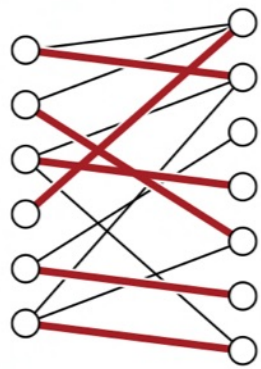


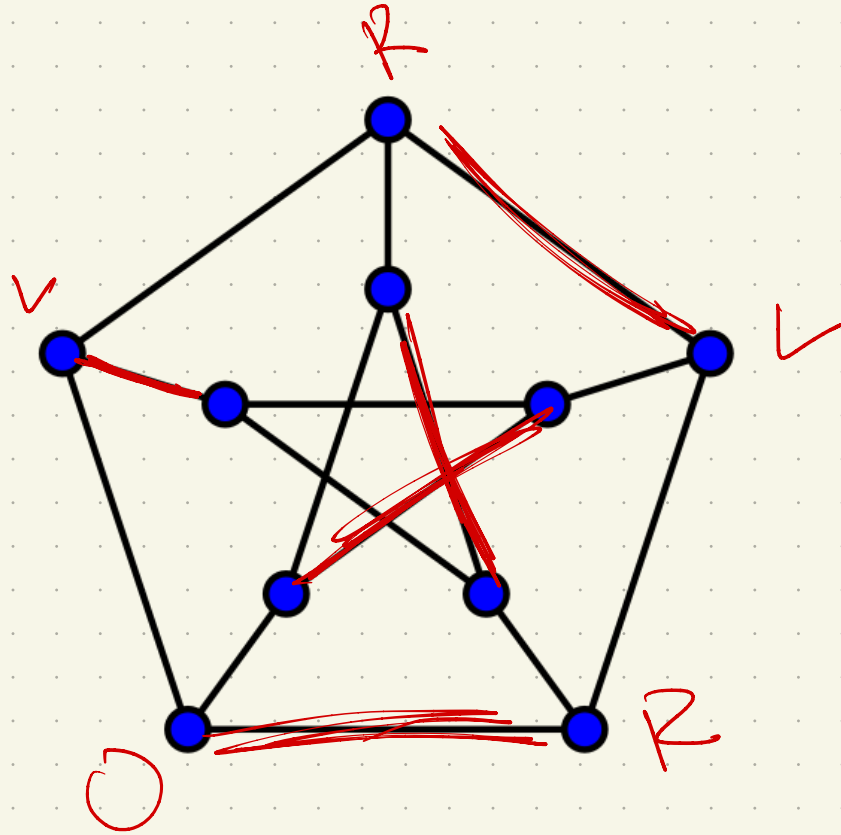
Figure 11.2. A maximum matching in a bipartite graph  $G$ , and the corresponding maximum flow in  $G'$ .

more general:



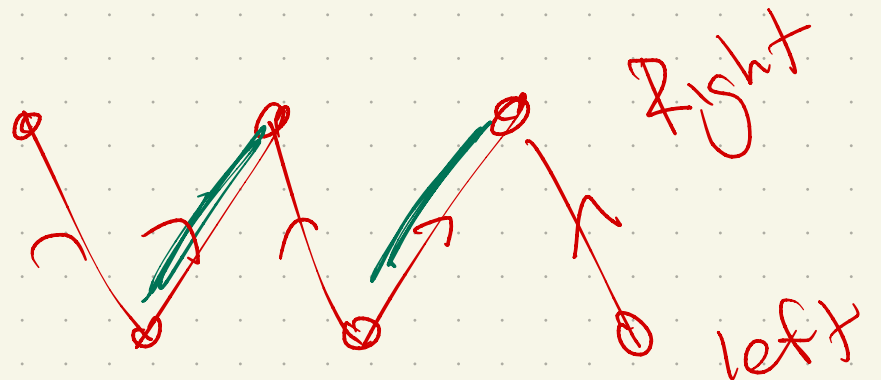
Why does this fail in general graphs?

First:  
maximum

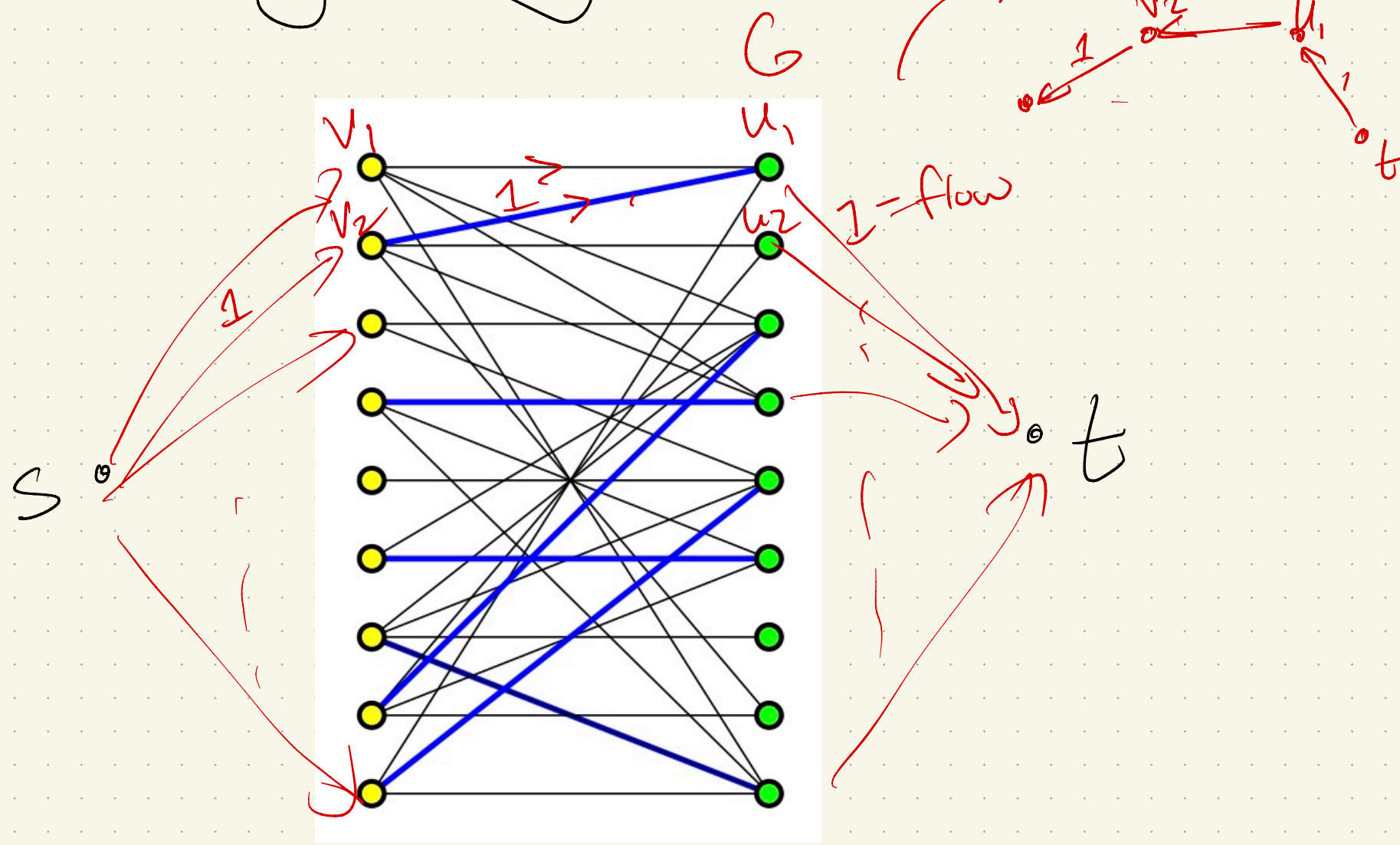


vs

maximal



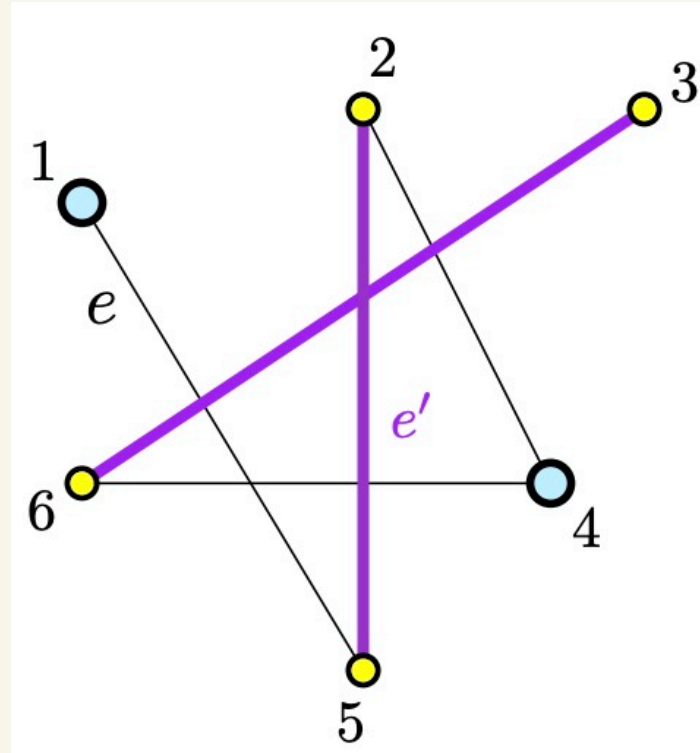
In bipartite graphs, residual networks find augmenting paths!



Similar techniques for general graphs  
(but no flows!)

Find an  
"alternating path":

Fix matching  $M$   
And a path that  
alternates edges  
in  $M$  & edges  
not in  $M$ .

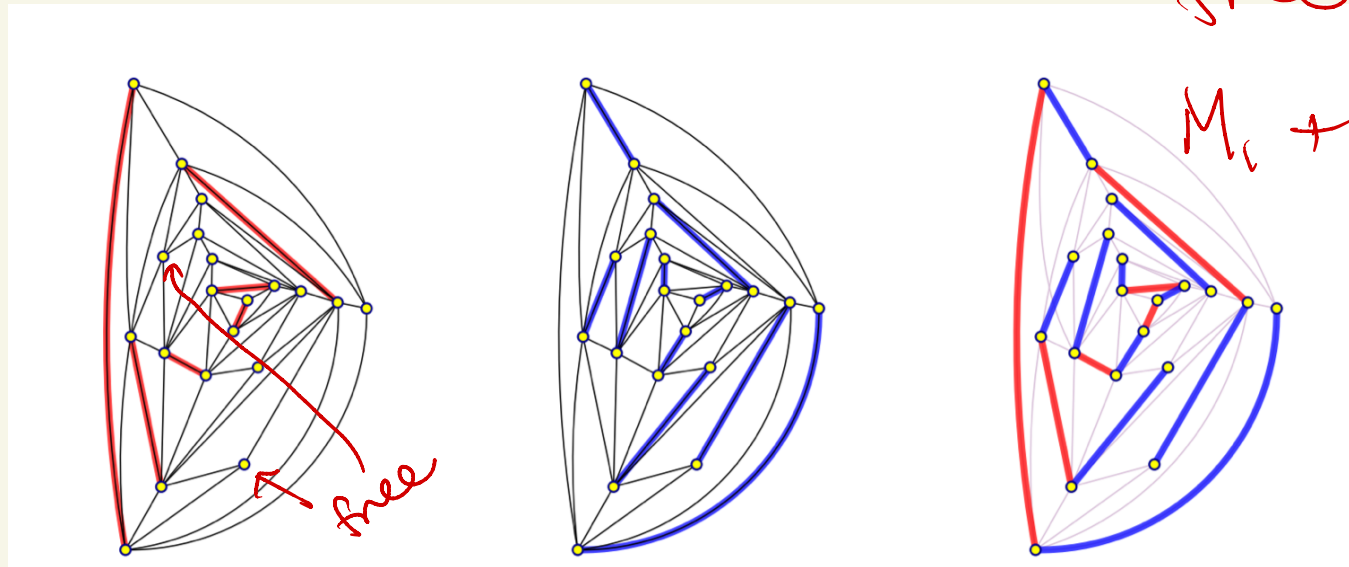


# Berge's Optimality Criteria:

A vertex is free if no edge from the matching is incident to it.

A matching  $M$  is optimum  $\iff$  no  $M$ -augmenting paths  
 $M$ -alternating paths starting & ending at free vertices

Why?

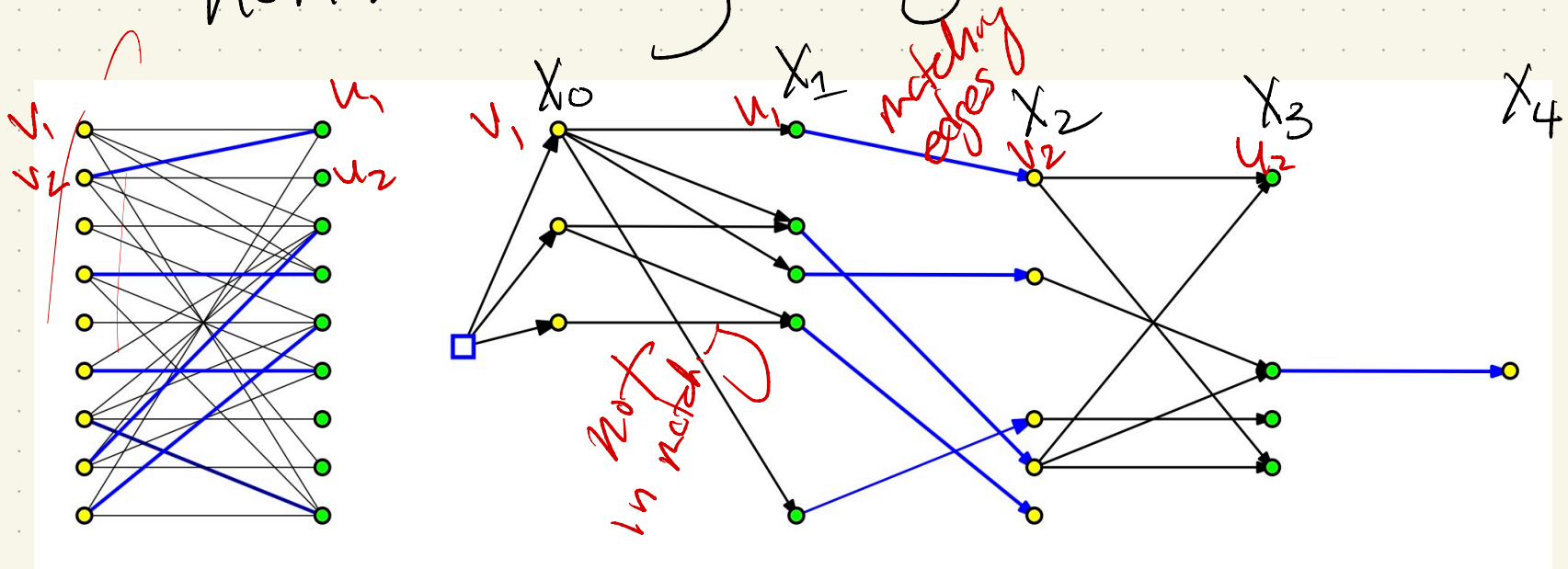


no  $\times$

Algorithm using modified BFS:

Take a matching, & start a BFS  
from open vertices in  $L$ :

Then alternate matching &  
non-matching edges in levels.



Goal: Find path from open  $v \in X_0$  to  
open  $w \in X_{2i+1}$   $\rightarrow$  bigger matching

Take away:

An algorithm.

modified BFS:  $O(V+E)$   
↳ find matching 1 larger

Runtime?  $O(V(V+E)) = O(VE)$

Rest of chapter: faster ways:  $O(E\sqrt{V})$   
 $O(E^{1+o(1)})$

General graphs:  $O(VE^2)$

↳ compute alternating paths